

DETAILED ACTION

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

Claims 1-97, 110, 114, 119, 123, 128-129 are rejected under 35 U.S.C. 102(e) as being anticipated by Kusterer et al. (US 2005/0076311A1) hereinafter Kusterer.

Claim 1: Kusterer teaches ***a method of configuring a server*** (e.g., portal server for the portal 320 as shown in Fig. 3, also see [0039] and [0042] “For example, the enterprise portal can reside in one or more servers connected to a public network and can run one or more server software programs”) ***to provide at least one composite user interface to a plurality of source applications*** (e.g., the user interface for the portal 320 as shown in Fig. 3 can be interpreted as the claimed “composite user interface”, a non-limiting and only an example presentation of which is shown in Fig. 5 and Fig.7. Kusterer mentions in [0041] “The integrated enterprise management system can consolidate and integrate the data and functionality of multiple different applications into a single enterprise management tool provided through the portal”. In other words, the portal 320 provides a composite

Art Unit: 2179

user interface for accessing data and functionality of the source applications 340 as shown in Fig. 3), ***the composite user interface comprising a plurality of user interface elements provided by the source applications*** (e.g., in the broadest reasonable interpretation, the phrase “*user interface elements*” refers to any “element” provided within the composite user interface. According to such interpretation, data and elements providing access to functionality, of multiple different applications provided within the integrated user interface of the portal (see [0041], as already mentioned above) reads on the limitations of the claim. Additionally, the portal presentations (see Fig. 5 and Fig. 7) taught by Kusterer include navigation areas (510, 520, 530 in Fig. 5 and 730, 740, 750 in Fig. 7) wherein “navigation nodes” providing access to the resources of the plurality of applications 340 are displayed. Referring to Fig. 5, Kusterer mentions, “These navigation iViews can present a united navigation hierarchy of navigation nodes in the form of a graphical hierarchical structure of expanding/collapsing containers and unit nodes, which can be used to launch application units in one or more other windows, which can be iViews. A target page window 540 present a work area that can display a launched navigation node.” See [0053]. With reference to the user interface displayed in Fig. 7, Kusterer further explains, “Navigation areas 730, 740, 750 provide a simple point and click user interface to the unified navigation hierarchy. A first navigation area 730 displays a graphical hierarchical structure of expanding/collapsing containers and

Art Unit: 2179

unit nodes, which can be used to launch application units.

Launching from one of the navigation areas can generate a client 'navigate' event which can be caught by the top level navigation which refreshes an inner page 760 with the URL of the launched

node". See [0059]. These navigation nodes which can be used to launch a plurality of

associated application units can also be interpreted as "*a plurality of user interface elements provided by the source applications*" because according to Kusterer these

nodes are derived from the source applications. See [0006] "Uniting the

navigation hierarchies can involve accepting connectors for the different application sources, and receiving the navigation

information from the different application sources through the connectors according to the navigation object model ...and

receiving the navigation information can involve receiving

navigation nodes." This clearly mentions that receiving navigation information

involves receiving navigation nodes from the different application sources. Additionally,

the one or more windows displaying the launched applications when their corresponding

navigation nodes are selected can also be interpreted as ""*a plurality of user interface*

elements provided by the source applications". This is because as already mentioned

above, Kusterer teaches that the navigation nodes "can be used to launch

application units in one or more other windows, which can be

iViews. A target page window 540 present a work area that can

display a launched navigation node." See [0053]. Therefore, although Fig. 5

Art Unit: 2179

and Fig. 7 shows only one target page window, Kusterer's invention is not limited to showing only one target page window, but can show the launched applications in a plurality of windows. One of ordinary skill in the art will also realize that the windows presenting a launched application will also contain user interface elements from the launched applications), ***the method comprising:***

combining a subset of the plurality of user interface elements from the source applications into composite user interface data (e.g., the composite user interface of the portal 320 as already mentioned above);

providing the composite user interface data to a user computer system for display as the composite user interface by the user computer system, wherein the composite user interface causes the user computer system to display the plurality of user interface elements provided by the source applications (e.g., displaying the user interface of the portal page);

receiving a user request from the user computer system relevant to at least one of the source applications (see, e.g., [0040] "The portal 320 receives requests from the clients 300 and generates information views 325 (e.g., Web pages) in response" and "The portal 320 receives information 345 from the applications 340 to fulfill requests from the clients 300". A user can select a navigation node at the user computer system relevant to at least one of the source applications and thereby access

Art Unit: 2179

informational content and functionality from said at least one of the source applications; see e.g., Fig. 7);

processing a model representing said composite user interface (e.g., the software implementing the composite portal interface can reasonably be interpreted in the broadest reasonable interpretation as “*a model representing said composite user interface*”, i.e., a software model of the portal interface) ***to generate rules for communication between said composite user interface and the source applications*** (see communicating with multiple application as mentioned in [0006], [0007], [0009], [0024], and [0026]. Such communication apparently involves generating rules for such communication); ***and***

generating one or more source requests to each relevant source application that represents the user request (e.g., as already mentioned above, a user can select a navigation node at the user computer system relevant to at least one of the source applications and thereby access informational content from said at least one of the source applications; see e.g., Fig. 7).

Claim 48 (a non-transitory program memory), Claim 49 (a computer apparatus), and Claim 80 (a server): are directed to the same invention of claim 1 in different statutory categories. Therefore, these claims are also rejected under similar rationale as claim 1.

Claim 2: Kusterer further teaches *a method according to claim 1, wherein said model* (i.e., the software implementing the composite portal interface) *comprises a model of at least part of a user interface provided by each source application* (see 600 and 620 in Fig. 6, that models at least part of a navigational hierarchy for a user interface, i.e., pages, provided by a source application) *and a model of relationships between the at least part of the user interface provided by each source application and the composite user interface* (e.g., 650 in Fig. 6 represents such modeling relationships between the pages provided by each source application and the composite user interface provided by the unified navigational interface application, i.e., the portal application).

Claim 81: is also rejected under the same rationale as claim 2 discussed hereinabove.

Claim 3: Kusterer further teaches *a method according to claim 1, further comprising:*

storing said rules within a hierarchical data structure comprising a plurality of entities (e.g., see the hierarchical structure mentioned in [0021], [0052], and [0053], also see Fig. 6, [0055] to [0057], Fig. 7 and [0059]).

Claim 82: is also rejected under the same rationale as claim 3 discussed hereinabove.

Claim 4: Kusterer further teaches *a method according to claim 3, further comprising: storing within said hierarchical data structure an entity representing the composite user interface* (e.g., the root node of the united navigation hierarchy, see also [0030], [0033]); *and associating with said entity a data group providing configuration data for the composite user interface* (e.g., associating configuration data in [0027], also role based configuration in [0005], [0010], [0037], and [0040]).

Claim 83: is also rejected under the same rationale as claim 4 discussed hereinabove.

Claim 5: Kusterer further teaches *a method according to claim 4, further comprising: storing within said hierarchical data structure a plurality of service entities representing processing modules which are together adapted to process user requests input to said composite user interface to produce one or more requests to at least one source application* (e.g., see nodes used to launch application units. See [0059]. Additionally, or in the alternative the interfaces can be interpreted as such service entities. See [0033]).

Claim 84: is also rejected under the same rationale as claim 5 discussed hereinabove.

Claim 6: Kusterer further teaches ***a method according to claim 5, wherein at least some of said service entities have an associated data group storing configuration data*** (e.g., properties of the node objects implicit in the reference).

Claim 85: is rejected under the same rationale as claim 6 discussed hereinabove.

Claim 7: Kusterer further teaches ***a method according to claim 6, wherein one of said service entities is an aggregation service entity representing an aggregation service configured to generate the one or more source application requests from the user request*** (e.g., see the navigation service as discussed in [0024]).

Claim 86: is rejected under the same rationale as claim 7 discussed hereinabove.

Claim 8: Kusterer further teaches *a method according to claim 7, wherein said aggregation service entity comprises: a child entity representing the composite user interface; and said child entity has at least one child entity representing one of the source applications* (see Fig. 6 and Fig. 7).

Claim 87: is rejected under the same rationale as claim 8 discussed hereinabove.

Claim 9: Kusterer further teaches *a method according to claim 3, wherein said rules are generated using a plurality of writers each writer being associated with an entity in said hierarchical data structure, and being adapted to write data to a data group associated with the respective entity* (e.g., see use of “connectors” as discussed in [0006], [0007], [0009], [0025] to [0027], and throughout the reference).

Claim 88: is rejected under the same rationale as claim 9 discussed hereinabove.

Claim 10: Kusterer further teaches *a method according to claim 9, wherein processing said model comprises:*

selecting one or more objects within said model;

determining one or more writers to be invoked to write data from the or each object to said hierarchical data structure; and

invoking the or each writer to write data to said hierarchical data structure (e.g., see use of “connectors” as discussed in [0006], [0007], [0009], [0025] to [0027], and throughout the reference).

Claim 89: is rejected under the same rationale as claim 10 discussed hereinabove.

Claim 11: Kusterer further teaches *a method according to claim 10, further comprising:*

determining from said at least one writer at least one further object within said model, and processing said further object (e.g., see use of “connectors” as discussed in [0006], [0007], [0009], [0025] to [0027], and throughout the reference).

Claim 90: is rejected under the same rationale as claim 11 discussed hereinabove.

Claim 12: Kusterer further teaches *a method according to claim 10, further comprising:*

identifying a further writer configured to identify an entity within said hierarchical data structure to which data is to be written (e.g., see use of “connectors” as discussed in [0006], [0007], [0009], [0025] to [0027], and throughout the reference).

Claim 91: is rejected under the same rationale as claim 12 discussed hereinabove.

Claim 13: Kusterer further teaches *a method according to claim 12, wherein said identifying an entity comprises:*

attempting to locate an entity within said hierarchical data structure to which data should be written; and

if said attempt is unsuccessful, creating an appropriate entity (e.g., see use of “connectors” as discussed in [0006], [0007], [0009], [0025] to [0027], and throughout the reference).

Claim 92: is rejected under the same rationale as claim 13 discussed hereinabove.

Claim 14: Kusterer further teaches ***a method according to claim 9, wherein each writer is a writer object which is an instance of a respective Java writer class*** (e.g., see use of “connectors” as discussed in [0006], [0007], [0009], [0025] to [0027], and throughout the reference).

Claim 93: is rejected under the same rationale as claim 14 discussed hereinabove.

Claim 15: Kusterer further teaches ***a method according to claim 14, wherein each writer class has a corresponding writer factory class*** (e.g., see use of “connectors” as discussed in [0006], [0007], [0009], [0025] to [0027], and throughout the reference).

Claim 94: is rejected under the same rationale as claim 15 discussed hereinabove.

Claim 16: Kusterer further teaches *a method according to claim 15, further comprising: registering each writer factory class with a writer lookup object; providing details of the or each object to be processed to said writer lookup object; and identifying one or more factory classes which should be used to create writer objects* (e.g., see use of “connectors” as discussed in [0006], [0007], [0009], [0025] to [0027], and throughout the reference).

Claim 95: is rejected under the same rationale as claim 16 discussed hereinabove.

Claim 17: Kusterer teaches *a method of generating model data representing a model of a composite user interface* (e.g., generating a navigational model data 650 as shown in Fig. 6 representing a model, i.e., a navigational model, of the unified application interface) *comprising a plurality of user interface elements provided by a plurality of source applications* (e.g., comprising, a plurality of user

interface elements, i.e., pages, as shown in navigation hierarchy 650 in Fig. 6 provided by a plurality of source applications), ***the method comprising:***

modelling at least part of a user interface provided by each of the source applications (see 600 and 620 in Fig. 6, that models at least part of a navigational hierarchy for the user interface, i.e., pages, provided by each of the source applications); ***and***

modelling relationships between the at least part of the user interfaces provided by the source applications and the composite user interface (e.g., 650 in Fig. 6 represents such modeling relationships between the pages provided by each source application and the composite user interface provided by the unified navigational interface application).

Claim 18: Kusterer further teaches ***a method according to claim 17, wherein the model is adapted for use in generating a composite application*** (e.g., see Abstract, [0006], [0007], [0009], [0024], and [0026]).

Claim 19: Kusterer further teaches ***a method according to claim 17, wherein modelling at least part of the user interface provided by the or each source application comprises:***

defining a plurality of source flow items each comprising a specified source user interface page provided by a source application; and

defining relationships between said plurality of source flow items (e.g., see Fig. 3, Fig. 6 and Fig. 7, wherein individual pages from applications can be interpreted as the source flow items).

Claim 20: Kusterer further teaches ***a method according to claim 19, wherein modelling at least part of the user interface provided by the or each source application further comprises:***

defining at least one page element within each specified source user interface page (e.g., defining a link or a URL. See [0007], [0023]).

Claim 21: Kusterer further teaches ***a method according to claim 19, wherein modelling at least part of the user interface provided by the or each source application further comprises: defining at least one flow control condition; associating a flow control condition with at least one of said plurality of source flow items*** (see [0023]).

Claim 22: Kusterer further teaches *a method according to claim 19, wherein modelling at least part of the user interface provided by the or each source application further comprises: defining request parameters used to obtain each specified source user interface page* (e.g., pages are obtained using links or URLs. See [0023], [0031], [0051], and [0059]).

Claim 23: Kusterer further teaches *a method according to claim 19, wherein modelling at least part of the user interface provided by the or each source application further comprises: defining at least one rule for each specified source user interface page which can be applied to enable recognition of the associated specified source user interface page* (e.g., pages are recognized using links or URLs. See [0023], [0031], [0051], and [0059]).

Claim 24: Kusterer further teaches *a method according to claim 23, wherein the or each rule is specified using a regular expression, or a path expression* (e.g., a link or URL apparently contains a path expression).

Claim 25: Kusterer further teaches *a method according to claim 17, wherein modelling at least part of the user interface provided by the or each source application further comprises: creating a plurality of objects which are*

instances of classes defined in an object oriented programming language (e.g., see Table 1 to Table 3).

Claim 26: Kusterer further teaches ***a method according to claim 17, wherein modelling relationships between the at least part of the user interface provided by the or each source application and the composite user interface comprises: combining at least part of a plurality of source application models*** (see Fig. 6).

Claim 27: Kusterer further teaches ***a method according to claim 17, further comprising: defining a plurality of composite flow items each comprising a specified user interface page; and defining relationships between said plurality of composite flow items*** (see Fig. 6 and Fig. 7).

Claim 28: Kusterer further teaches ***a method according to claim 19, further comprising: defining a plurality of composite flow items each comprising a specified user interface page; and defining relationships between said plurality of composite flow items, wherein at least one composite flow item is a source flow item, and said specified user interface page is a specified source user interface page*** (see Fig. 6 and Fig. 7).

Claim 29: Kusterer further teaches *a method according to claim 27, wherein at least one specified user interface page is a composite user interface page* (see Fig. 7).

Claim 30: Kusterer further teaches *a method according to claim 20, wherein at least one specified user interface page is a composite user interface page, the method further comprising: modelling manipulations which are applied to said at least one page element within a specified source user interface page to create said composite user interface page* (e.g., see Fig. 7 wherein the pages accessed is a composite source application page and the elements of the page is manipulated within the unified interface of Fig. 7 similar to the manipulation in the source application).

Claim 31: Kusterer further teaches *a method according to claim 30, further comprising: specifying an ordered plurality of manipulations to be carried out to create said composite user interface page* (e.g., displaying the composite user interface page of Fig. 7 using software instruction executed in a specified order by the processor).

Claim 32: Kusterer further teaches ***a method according to claim 17, further comprising modelling at least one further user interface element to be included in the composite user interface*** (see Fig. 6 and Fig. 7, which shows modeling an arbitrary number of source application's user interface elements in the composite user interface).

Claims 33-47: recite similar limitations as claims 1-16 respectively, and therefore rejected under similar rationale as discussed in the rejections of claims 1-16 hereinabove.

Claims 50-79: are directed to an apparatus implementing the method of claim 17, and 19-47 respectively, and therefore rejected under similar rationale as discussed in the rejections of claims 17 and 19-47.

Claim 96:

A computer apparatus for generating a composite user interface for communication with a plurality of source applications, the apparatus comprising:

a program memory that includes processor readable instructions (e.g., see [0063]);

a processor for reading and executing the instructions contained in the program memory (see e.g., [0063]), ***wherein the instructions are for:***

generating model data representing a model of said composite user interface in response to user input (e.g., generating a navigational model data 650 as shown in Fig. 6 representing a model, i.e., a navigational model, of the unified application interface), ***wherein the composite user interface comprises a plurality of user interface elements provided by the source applications*** (e.g., in the broadest reasonable interpretation, the phrase “*user interface elements*” refers to any “element” provided within the composite user interface. According to such interpretation, data and elements providing access to functionality, of multiple different applications provided within the integrated user interface of the portal (see [0041], as already mentioned above) reads on the limitations of the claim. Additionally, the portal presentations (see Fig. 5 and Fig. 7) taught by Kusterer include navigation areas (510, 520, 530 in Fig. 5 and 730, 740, 750 in Fig. 7) wherein “navigation nodes” providing access to the resources of the plurality of applications 340 are displayed. Referring to Fig. 5, Kusterer mentions, “These navigation iViews can present a united navigation hierarchy of navigation nodes in the form of a graphical hierarchical structure of expanding/collapsing containers and unit nodes, which can be used to launch application units in one or more other windows, which can be iViews. A target page window 540 present a work area that can display a launched navigation node.” See [0053]. With reference to the user interface displayed in Fig. 7, Kusterer

Art Unit: 2179

further explains, "Navigation areas 730, 740, 750 provide a simple point and click user interface to the unified navigation hierarchy. A first navigation area 730 displays a graphical hierarchical structure of expanding/collapsing containers and unit nodes, which can be used to launch application units. Launching from one of the navigation areas can generate a client 'navigate' event which can be caught by the top level navigation which refreshes an inner page 760 with the URL of the launched node". See [0059]. These navigation nodes which can be used to launch a plurality of associated application units can also be interpreted as "*a plurality of user interface elements provided by the source applications*" because according to Kusterer these nodes are derived from the source applications. See [0006] "Uniting the navigation hierarchies can involve accepting connectors for the different application sources, and receiving the navigation information from the different application sources through the connectors according to the navigation object model ...and receiving the navigation information can involve receiving navigation nodes." This clearly mentions that receiving navigation information involves receiving navigation nodes from the different application sources. Additionally, the one or more windows displaying the launched applications when their corresponding navigation nodes are selected can also be interpreted as ""*a plurality of user interface elements provided by the source applications*". This is because as already mentioned

above, Kusterer teaches that the navigation nodes “can be used to launch application units in one or more other windows, which can be iViews. A target page window 540 present a work area that can display a launched navigation node.” See [0053]. Therefore, although Fig. 5 and Fig. 7 shows only one target page window, Kusterer’s invention is not limited to showing only one target page window, but can show the launched applications in a plurality of windows. One of ordinary skill in the art can appreciate that the windows presenting a launched application can also contain user interface elements from the launched application) ;

storing said model (see e.g., [0030] “The navigation service can store navigation information in the form of navigation nodes 220, 224 in a data structure representing a united navigation hierarchy”);

reading said model from said memory and generating a configuration data structure (i.e., reading the navigational model from memory is implied as displaying the nodes on the display according to the navigational hierarchy requires reading the hierarchy data from memory. Additionally, generating a configuration data structure is also taught as “personalization” capability, since generating and storing personalization data can be interpreted as generating configuration data structure);

receiving a request from a composite user interface (see, e.g., [0040] “The portal 320 receives requests from the clients 300 and generates information views 325 (e.g., Web pages) in response” and “The portal 320 receives information 345 from the

applications 340 to fulfill requests from the clients 300". A user can select a navigation node at the user computer system relevant to at least one of the source applications and thereby access informational content from said at least one of the source applications; see e.g., Fig. 7);

generating a source application request to at least one of said source application in response to said request, in accordance with data stored in said configuration data structure (e.g., as already mentioned above, a user can select a navigation node at the user computer system relevant to at least one of the source applications and thereby access informational content from said at least one of the source applications; see e.g., Fig. 7); ***and***

transmitting said source application request to said at least one of said source applications (as already discussed above, accessing informational content from source applications require transmitting the information request to the source application).

Claim 97:

This claim recites similar limitations as claim 17 except for the limitation reciting "*generating a source application model for each of the at least one source applications*". Kusterer teaches generating source application models for each of the source applications as source hierarchies as illustrated in Fig. 6 and also discussed in [0004],

[0016], and [0010] and throughout the reference. Therefore, this claim is also rejected under the same rationale as discussed in the rejection of claim 17 hereinabove.

Claim 110: Kusterer teaches ***a method for generating a composite user interface*** (e.g., a portal interface as illustrated in Fig. 5) ***comprising a plurality of user interface elements provided by at least one source application*** (e.g., comprising a plurality of user interface elements, i.e., pages, provided by at least one source application as illustrated in Fig. 6), ***the method comprising:***

combining a subset of the plurality of user interface elements from the source applications into composite user interface data (e.g., referring to the portal presentation of Fig. 5, the portal presentation 500 at a given point in time can be interpreted as a “*composite user interface data*” generated from combining a subset of the plurality of user interface elements from the source applications); ***and***

selecting said composite user interface from a plurality of predefined composite user interfaces on the basis of at least one predefined parameter (e.g., in the basis of user's role. See [0005], [0010], [0037], and [0040]).

Claim 119, 128, and 129 are rejected under the same rationale as claim 110 over Kusterer as discussed hereinabove.

Claim 114: Kusterer further teaches ***a method according to claim 110, further comprising:***

receiving at least one request message generated by said composite user interface (e.g., a user's selection event in the unified user interface requesting a resource from a source application);

producing at least one further message in response to said request message (e.g., a message generated by Applications Connectors to be sent to the appropriate application associated with the interface element selected by the user); ***and***

forwarding said at least one further message to one of said at least one source applications (e.g., the Application Connector forwards the request message to the application for processing. See [0009], [0024], [0045] and throughout the reference).

Claim 123: is also rejected under the same rationale as claim 114 over Kusterer as discussed hereinabove.

Claims 110-111, 114, 118-120, 123, and 127-129 are rejected under 35 U.S.C. 102(b) as being anticipated by Gangopadhyay (US 2002/0184402 A1).

Claim 110: Gangopadhyay teaches ***a method for generating a composite user interface*** (e.g., a Personal Workflow with associated services as illustrated in Table 2) ***comprising a plurality of user interface elements provided by at least one source application*** (e.g., the Services as illustrated in Table 2), ***the method comprising: combining a subset of the plurality of user interface elements from the source applications into composite user interface data*** (i.e., a “user interface element” is interpreted as a menu option for a service as shown in the “Services” column in Table 2. These services are provided by respective source applications, see [0068], [0084], [0092]. Therefore, these menu options for services provided by respective source applications can be reasonably interpreted, in the broadest reasonable interpretation, as user interface elements provided by the source applications); ***and selecting said composite user interface from a plurality of predefined composite user interfaces on the basis of at least one predefined parameter*** (e.g., since the Personal Workflow is a series of potential or planned events listed in a time ordered fashion and may take the form of a business traveler’s Appointment Calendar, it is implicit in the reference that a particular Personal Workflow can be chosen by the user based on date or time from the calendar).

Claim 111: Gangopadhyay further teaches *a method according to claim 110, wherein said at least one predefined parameter comprises a parameter relating to at least one of time of day and date* (as already discussed in the rejection of claim 110 hereinabove).

Claim 114: Gangopadhyay further teaches *a method according to claim 110, further comprising:*

receiving at least one request message generated by said composite user interface;

producing at least one further message in response to said request message; and

forwarding said at least one further message to one of said at least one source applications (e.g., see Context Server's Processing of a Request for Application Services as discussed in section 7.3.4. See [0108] to [0116]).

Claim 118: Gangopadhyay further teaches *a method according to claim 110, wherein at least two of said plurality of predefined composite user interfaces comprise different source user interface elements* (e.g., apparently, two different Personal Workflow interface will show different user interface elements).

Independent claims 119 (an apparatus), 128 (a non-transitory program memory) and 129 (a computer) are directed to the same invention of claim 110 in different form (i.e., subject matter). Therefore, these forms are either implicitly or explicitly taught by the Gangopadhyay reference.

Dependent claims 120, 123, and 127: recite similar limitations as claims 111, 114, and 118 respectively. Therefore these claims are rejected under similar rationale as discussed in the rejection of claims 111, 114, and 118 respectively hereinabove.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was

Art Unit: 2179

not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

Claims 98-109 are rejected under 35 U.S.C. 103(a) as being unpatentable over Gangopadhyay (US 2002/0184402 A1) hereinafter Gangopadhyay in view of Kobipalayam Murugaiyan (US 2004/0168122 A1) hereinafter Murugaiyan.

Claim 98: Gangopadhyay teaches *a method for providing a composite user interface comprising a plurality of user interface elements provided by at least one source application* (e.g., providing a user interface as shown in Table 2), *the method comprising:*

monitoring operation of the composite user interface to obtain management data (e.g., monitoring the interface to obtain user selection data and/or usage context data, i.e., interpreted as management data, since the user selection data and/or usage context data is used to manage changes made to the interface), *wherein the management data includes usage conditions* (e.g. usage context); *and*

modifying said composite user interface by changing a number of the user interface elements for display by a user computer system in accordance with the usage conditions (see e.g., Gangopadhyay dynamically constructs menus of services relevant to any usage context. See Abstract. Therefore, clearly, when the usage context

Art Unit: 2179

changes, the number of relevant services displayed also changes, i.e., the composite user interface is dynamically modified by changing a number of the services relevant to the current usage context).

Gangopadhyay does not teach ***wherein the usage conditions include demand for information and response time for providing the demanded information.*** However, Murugaiyan teaches a technique of rendering a requested web page with fast response time by partially rendering the web page using available page data and later completing the rendering process by rendering dynamic data when that dynamic data cannot be obtained quickly. See e.g. [0004], [0005] and [0030]. Therefore, it would have been obvious to one of ordinary skill in the art to utilize this known technique taught by Murugaiyan with the invention as disclosed by Gangopadhyay. To do this, it would have been obvious to one of ordinary skill in the art to monitor usage conditions including response time for providing demanded information and displaying static information first with later rendition of dynamic information if the response time for the dynamic information is longer than an acceptable threshold. Such a modification is considered the result not of novelty but of ordinary skill and common sense.

Claim 99: Gangopadhyay further teaches ***a method according to claim 98, wherein the user interface elements comprise mandatory and nonmandatory user interface elements, the method further comprising modifying said composite user***

interface in response to said management data to (i) display the nonmandatory user interface elements during a first usage condition and (ii) not display the mandatory user interface elements during a second usage condition, wherein the first usage condition represents a lower usage than the second usage condition (see e.g., cancelling an event and thereby updating the workflow interface based on the cancelled event. See [0118] to [0121]).

Claim 100: Gangopadhyay further teaches ***a method according to claim 99, wherein said modifying comprises deleting some of said plurality of user interface elements from said composite user interface*** (see [0121]).

Claim 101: Gangopadhyay teaches all the limitations of the claim as recited in claim 98, except the limitation reciting ***further comprising producing data representing usage patterns of said composite user interface using said management data***. However, the Examiner takes official notice that it was well-known in the art to monitor user's usage of an interface to produce usage data in order to determine usage pattern and provide customized user interface based on such usage pattern. Therefore, it would have been obvious to one of ordinary skill in the art to modify the Personal Workflow interface as taught in Gangopadhyay to provide customized messages, advertisements or other user interface elements based on

Art Unit: 2179

usage pattern as claimed. Such modification would have been the result not of novelty but of ordinary skill in the art.

Claim 102: Gangopadhyay further teaches *a method according to claim 98, further comprising: receiving at least one request message generated by said composite user interface; producing at least one further message in response to said request message; and forwarding said at least one further message to one of said at least one source applications* (e.g., notifying participants of an event upon cancellation of the event as discussed in [0043]).

Claims 103 (a non-transitory program memory), 104 (a computer apparatus), and 105 (an apparatus) are directed to the same invention of claim 98 as different statutory subject-matter categories. These categories are either implicitly or explicitly taught by the Gangopadhyay reference. Therefore, these claims are also rejected under similar rationale as claim 98.

Dependent claims 106-109: recite similar limitations as claims 99-102 respectively. Therefore, these claims are rejected under similar rationale as claims 99-102 respectively as discussed in detail hereinabove.

Claims 112-113, 115-117, 121-122, and 124-126 are rejected under 35 U.S.C. 103(a) as being unpatentable over Kusterer.

For claims 112-113 and 115-117, the Examiner took official notice that the limitations recited in these claims (e.g., selecting composite user interface based on a parameter relating to usage statistics as recited in claim 112, and selecting composite user interface based on a parameter relating to a marketing campaign as recited in claim 113, pre-rendering a portal page when only the mandatory elements are received and while waiting for the non-mandatory elements as recited in claim 115, having identical source user interface elements as recited in claim 116, and having different mandatory user interface elements as recited in claim 117) were well-known in the art at the time of the invention. For example, it was well-known to provide different portal interface to a user based on usage statistics or based on different marketing campaign as recited in claims 112 and 113 respectively. It is also well-known to pre-render a web-page when some essential items have been retrieved over the network and some non-essential items are not yet received as recited in claim 115. Therefore, it would have been part of the ordinary capabilities of a person skilled in the art to implement these techniques in the portal interface taught by Kusterer and thereby arrive at the present invention. Such modifications would have been the result not of innovation but of ordinary skill and common sense. Applicant did not challenge the official notice in the response.

Claims 121-122 and 124-126: recite similar limitations as claims 112-113 and 115-117 respectively and therefore rejected under the same rationale discussed hereinabove.

Response to Arguments

Applicant's arguments filed 7/6/2011 have been fully considered but they are not persuasive.

Claims 1, 80, 110, 119: Applicants argued that Kusterer does not teach the limitation “*combining a subset of the plurality of user interface elements from the source applications into composite user interface data*”. See pages 29-30 in the Remarks. The Examiner disagrees. In the broadest reasonable interpretation, the phrase “*user interface elements*” refers to any “element” provided within the composite user interface. According to such interpretation, data and elements providing access to functionality, of multiple different applications provided within the integrated user interface of the portal (see [0041]) reads on the limitations of the claim. Additionally, the portal presentations (see Fig. 5 and Fig. 7) taught by Kusterer include navigation areas (510, 520, 530 in Fig. 5 and 730, 740, 750 in Fig. 7) wherein “navigation nodes” providing access to the resources of the plurality of applications 340 are displayed. Referring to Fig. 5, Kusterer mentions, “These navigation iViews can present a united navigation hierarchy of navigation nodes in the form of a graphical hierarchical structure of expanding/collapsing containers and unit nodes, which can be used to launch application units in one

Art Unit: 2179

or more other windows, which can be iViews. A target page window 540 present a work area that can display a launched navigation node." See [0053]. With reference to the user interface displayed in Fig. 7, Kusterer further explains, "Navigation areas 730, 740, 750 provide a simple point and click user interface to the unified navigation hierarchy. A first navigation area 730 displays a graphical hierarchical structure of expanding/collapsing containers and unit nodes, which can be used to launch application units. Launching from one of the navigation areas can generate a client 'navigate' event which can be caught by the top level navigation which refreshes an inner page 760 with the URL of the launched node". See [0059]. These navigation nodes which can be used to launch a plurality of associated application units can also be interpreted as "*a plurality of user interface elements provided by the source applications*" because according to Kusterer these nodes are derived from the source applications. See [0006] "Uniting the navigation hierarchies can involve accepting connectors for the different application sources, and receiving the navigation information from the different application sources through the connectors according to the navigation object model ...and receiving the navigation information can involve receiving navigation nodes." This clearly mentions that receiving navigation information involves receiving navigation nodes from the different application sources. Additionally,

Art Unit: 2179

the one or more windows displaying the launched applications when their corresponding navigation nodes are selected can also be interpreted as *"a plurality of user interface elements provided by the source applications"*. This is because as already mentioned above, Kusterer teaches that the navigation nodes "can be used to launch application units in one or more other windows, which can be iViews. A target page window 540 present a work area that can display a launched navigation node." See [0053]. Therefore, although Fig. 5 and Fig. 7 shows only one target page window, Kusterer's invention is not limited to showing only one target page window, but can show the launched applications in a plurality of windows. One of ordinary skill in the art can appreciate that the windows presenting a launched application can also contain user interface elements from the launched applications.

Claim 17: besides the arguments which have already been addressed with regard to claim 1, 80, 110 and 119, Applicants additionally argued that the navigation service and navigation hierarchies are not "modeling relationships between the at least part of the user interface provided by the source application and the composite user interface" as required by claim 17. However, this amounts to a mere assertion and Applicants failed to specifically point out how the cited portions of the reference support Applicants' assertion. Therefore, the Examiner disagrees with Applicants' assertion and maintains the rejection based on the view that the mapping of the nodes from individual applications into the nodes of the integrated hierarchy constitutes modelling navigation

Art Unit: 2179

relationships between the at least part of the user interfaces provided by the source applications and the composite user interface as claimed.

Claim 50, 96: in response to Applicants arguments with respect to claims 50 and 96 (see pages 31-32), the same response as provided for claim 17 above applies.

Claims 98 and 105: Applicant's arguments with respect to claims 98 and 105 (see page 32) have been considered but are moot in view of the new ground(s) of rejection.

Claim 110 and 119: Applicants argued that claims 110 and 119 require that the actual user interface elements themselves are from the source applications and therefore Gangopadhyay does not teach or suggest that the actual user interface elements themselves are from the source applications. See page 33 in the Remarks. The Examiner disagrees. The Examiner points out that Applicants failed to point out what is meant by "actual user interface elements" or failed to clarify the scope of the term "user interface elements". In the broadest reasonable interpretation , the phrase "*user interface elements*" refers to any "element" provided within a user interface. According to such interpretation, data and elements providing access to functionality of a source application is considered to be a user interface element from the source application. Therefore, Gangopadhyay teaches the limitations according to the arguments already provided in the previous office action.

Art Unit: 2179

For rest of the claims, Applicants relied on the same arguments that have been addressed in the response above.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to RASHEDUL HASSAN whose telephone number is (571)272-9481. The examiner can normally be reached on M-F 7:30AM - 4PM EST.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Weilun Lo can be reached on 571-272-4847. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Rashedul Hassan/
Examiner, Art Unit 2179

/Weilun Lo/
Supervisory Patent Examiner, Art Unit 2179